Apéndice C

Códigos Auxiliares

En este apéndice se muestran programas realizados en C/C++ como herramientas que ayudan a para mostrar de mejor manera los resultados buscados por la investigación de tesis. Para el primer caso se realiza un programa que muestra las tablas a partir de los modelos *Answers Sets* encontrados por Clasp y para Prover9 se realiza un programa para mostrar la prueba encontrada bajo la notación habitual.

C.1. Parser de Clasp

```
#include <stdio.h>
#define NUM 3

int i, j;
char s[1000];
int impl[NUM][NUM],
    and[NUM][NUM],
    or[NUM][NUM],
    neg[NUM];

void Print2DTables(int T[NUM][NUM],
                   int T2[NUM][NUM],
                   int T3[NUM][NUM]){
  for(i = 0; i < NUM; i++)
    for(j = 0; j < NUM; j++)
      printf("%d %d   %d     %d    %d\n",
```

```
17          i, j, T[i][j], T2[i][j], T3[i][j]);
18  }
19
20  void Print1DTables(int T[NUM]){
21    for(i = 0; i < NUM; i++)
22      printf("%d  %d\n", i, T[i]);
23  }
24
25  int main(){
26    while(scanf("%s", s) && s[0] != 'S'){
27      if(s[0] == 'i')
28        impl[s[5] - 48][s[7] - 48] = s[9] - 48;
29      if(s[0] == 'a')
30        and[s[4] - 48][s[6] - 48] = s[8] - 48;
31      if(s[0] == 'o')
32        or[s[3] - 48][s[5] - 48] = s[7] - 48;
33      if(s[0] == 'n')
34        neg[s[4] - 48] = s[6] - 48;
35    }
36
37    printf("a b a->b a^b avb\n");
38    Print2DTables(impl, and, or);
39    printf("a ~a\n");
40    Print1DTables(neg);
41
42    return 0;
43  }
```

```
➜ Avance de Tesis  gcc -o translateClingo translateClingo.c
➜ Avance de Tesis  clingo C1 | ./translateClingo
a b a->b a^b avb
0 0  2   0   0
0 1  1   0   1
0 2  2   0   2
1 0  0   0   1
1 1  1   2   1
1 2  1   1   1
2 0  0   0   2
2 1  1   1   1
2 2  2   2   2
a ~a
0  2
1  1
2  0
```

Figura 1. Ejecución del Parser para Clasp.

C.2. Parser de Prover9

```
#include <iostream>
#include <stack>
#include <stdio.h>
using namespace std;

int main(){

  stack <string> S;
  string input, left, right;
  int i, j, l;
  char s[1000];

  while(cin >> input){
    if(input[0] == 'P' && input[1] == '(' ){
      l = input.length(), j = 0;
      for(i = l - 1; i >= 0; i--)
        if(input[i] != '(' &&
           input[i] != ')' &&
           input[i] != '.' &&
           input[i] != ',' &&
           input[i] != 'P'){
          if(input[i] == 'l') i -= 3;
          if(input[i] == 'r') i -= 1;
          if(input[i] == 'd' || input[i] == 'g') i -= 2;
          s[j++] = input[i];
        }
      for(i = 0; i < j; i++){
        if(s[i] == 'i') {
          right = S.top();
          S.pop();
          left = S.top();
          S.pop();
          S.push("(" + left + " <- " + right + ")");
        }
        else{
          if(s[i] == 'o') {
            right = S.top();
```

```
38            S.pop();
39            left = S.top();
40            S.pop();
41            S.push("(" + left + " v " + right + ")");
42          }
43          else{
44            if(s[i] == 'a') {
45              right = S.top();
46              S.pop();
47              left = S.top();
48              S.pop();
49              S.push("(" + left + " ^ " + right + ")");
50            }
51            else{
52              if(s[i] == 'n') {
53                left = S.top();
54                S.pop();
55                S.push("(" + left + "~)");
56              }
57              else
58                S.push(string(1, s[i]));
59            }
60          }
61        }
62      }
63      input = S.top();
64      j = input.length();
65      for(i = j - 1; i >= 0; i--){
66        if(input[i] == '(') printf(")");
67        else{
68          if(input[i] == ')') printf("(");
69          else{
70            if(input[i] == '<') printf(">");
71            else
72              printf("%c", input[i]);
73          }
74        }
75      }
76      input = "";
```

```
77        printf(" ");
78        S.pop();
79
80      }
81      if(input[0] == '[')
82        cout << input << endl;
83      if(input[0] <= 48 || input[0] <= 57)
84        cout << input << " ";
85    }
86    return 0;
87  }
```

```
➜ Avance de Tesis  g++ -o translateProver9 translateProver9.cpp
➜ Avance de Tesis  prover9 -f C1.in | prooftrans | ./translateProver9
(64) 2009-11A, 2009. 1196 26 21:08:25 2014 "prover9 -f -------- Proof 1 --------


THEOREM PROVED

------ process 1196 exit (max_proofs) ------
% -------- -------- % 1 0.11 (+ 0.01) % 9. % 3. % 14.000. % 207. 1 (x -> x)  # #
 [goal].
2 (x -> (y -> x))  [assumption].
3 ((x -> y) -> ((x -> (y -> z)) -> (x -> z)))  [assumption].
8 (x -> (y v x))  [assumption].
16 -P(impl(x,y)) -P(x) y  [assumption].
17 -P(impl(c1,c1)). [deny(1)].
79 ((x -> ((y v x) -> z)) -> (x -> z))  [hyper(16,a,3,a,b,8,a)].
4585 (x -> x)  [hyper(16,a,79,a,b,2,a)].
4586 $F. [resolve(4585,a,17,a)].
```

Figura 2. Ejecución del Parser para Prover9.